

AMENDMENTS TO THE CLAIMS

1-12. (Cancelled)

13. (Currently Amended) A computer-implemented method for generating a message structure for a message for making an application program interface (API) call in a graphical user interface, the method comprising:

displaying a representation for a root node in response to a request to display a new message structure, where the representation appears in a pane of the graphical user interface;

receiving a request to add a child node to the root node, where the child node is selected from the group consisting of element nodes, field nodes, method nodes, and parameter nodes;

displaying a representation of the added child node to the root node such that a hierarchical relationship between the added child node and the root node is illustrated in a hierarchical tree; ~~and~~

automatically generating code for the message structure, wherein the message structure is represented by the hierarchical tree, wherein the code includes embedded delimiters that indicate a hierarchical structure of the message structure, and wherein the message structure corresponds to a message for making ~~the an~~ API ~~calls~~ call;

detecting that a pointer that is manipulated by a pointing device is guided over a portion of a displayed node in the hierarchical tree;

receiving an indication that a button on the pointing device has been selected and continues to be selected when the pointer is over the portion of the displayed node;

detecting movement of the pointer by the pointing device;

visually dragging the displayed node in correspondence with the movement of the pointer;

receiving an indication that the button on the pointing device has been released;

detecting that the pointer has moved from an original point in the hierarchical tree to a second point in the hierarchical tree, where the second point corresponds to a location of the pointer when the button has been released;

dropping the displayed node at the second point to display a revised hierarchical tree; and

automatically regenerating code for the message structure, wherein the message structure is represented by the revised hierarchical tree, and wherein the code includes embedded delimiters that indicate the hierarchical structure of the message structure represented by the revised hierarchical tree.

14. (Previously Presented) The method as defined in Claim 13, further comprising:  
receiving a request to add a sub-node to the child node; and  
displaying a representation of the added sub-node such that the hierarchical relationship between the added sub-node and the child node is illustrated in the hierarchical tree.

15. Canceled

16. (Currently Amended) The method as defined in Claim 13, further comprising:

detecting that a pointer is over a portion of a displayed node in the hierarchical tree and that an input from a pointing device has been received such that the displayed node is selected; and

displaying a list of properties for the selected node in a separate pane of the graphical user interface in response to the selection of the node.

17. (Original) The method as defined in Claim 13, wherein the child node that is added is selected from a pre-programmed API function.

18. (Original) The method as defined in Claim 13, wherein the child node that is added is selected from a pre-programmed interface method.

19. (Currently Amended) A computer-readable medium having computer-executable instructions stored thereon which, when executed by one or more processors, cause the one or more processors to perform ~~for performing~~ a method for generating a message structure for a message for making an application program interface (API) call, the method comprising:

displaying a representation for a root node in response to a request for a new message structure, where the representation appears in a pane of the graphical user interface;

receiving a request to add a child node to the root node, where the child node is selected from the group consisting of element nodes, field nodes, method nodes, and parameter nodes;

displaying a representation of the added child node to the root node such that a hierarchical relationship between the added child node and the root node is illustrated in a hierarchical tree; ~~and~~

automatically generating code for the message structure, wherein the message structure is represented by the hierarchical tree, wherein the code includes embedded delimiters that indicate a hierarchical structure of the message structure, and wherein the message structure corresponds to a message for making ~~the~~ an API calls call;

detecting that a pointer that is manipulated by a pointing device is guided over a portion of a displayed node in the hierarchical tree;

receiving an indication that a button on the pointing device has been selected and continues to be selected when the pointer is over the portion of the displayed node;

detecting movement of the pointer by the pointing device;

visually dragging the displayed node in correspondence with the movement of the pointer;

receiving an indication that the button on the pointing device has been released;

detecting that the pointer has moved from an original point in the hierarchical tree to a second point in the hierarchical tree, where the second point corresponds to a location of the pointer when the button has been released;

dropping the displayed node at the second point to display a revised hierarchical tree; and

automatically regenerating code for the message structure, wherein the message structure is represented by the revised hierarchical tree, and wherein the code includes embedded delimiters that indicate the hierarchical structure of the message structure represented by the revised hierarchical tree.

20. (Currently Amended) A system for generating a message structure for a message for making an application program interface (API) call, the system comprising:

a graphical user interface;

a module configured to display a representation for a root node in response to a request for a new message structure, where the representation appears in a pane of the graphical user interface;

a module configured to receive a request to add a child node to the root node, where the child node is selected from the group consisting of element nodes, field nodes, method nodes, and parameter nodes;

a module configured to display a representation of the added child node to the root node such that a hierarchical relationship between the added child node and the root node is illustrated in a hierarchical tree; ~~and~~

a module configured to automatically generate code for the message structure, wherein the message structure is represented by the hierarchical tree, wherein the code includes embedded delimiters that indicate a hierarchical structure of the message structure, and wherein the message structure corresponds to a message for making ~~the an~~ API calls call;

a module configured to detect when a pointer that is manipulated by a pointing device is guided over a portion of a displayed node in the hierarchical tree;

a module configured to detect that a button on the pointing device has been selected and continues to be selected when the pointer is over the potion of the displayed node;

a module configured to detect a position of the pointer;

a module configured to visually drag the displayed node in the graphical user interface in correspondence with motion of the pointer;

a module configured to receive an indication that the button on the pointing device has been released;

a module configured to detect that the pointer has moved from an original point in the hierarchical tree to a second point in the hierarchical tree, where the second point corresponds to a location of the pointer when the button has been released;

a module configured to drop the displayed node at the second point to display a revised hierarchical tree; and

a module configured to automatically regenerate code for the message structure, wherein the message structure is represented by the revised hierarchical tree, and wherein the code includes embedded delimiters that indicate the hierarchical structure of the message structure represented by the revised hierarchical tree.

21-50. (Cancelled)

51. (New) The computer readable medium as defined in Claim 19, wherein the method further comprises:

receiving a request to add a sub-node to the child node; and  
displaying a representation of the added sub-node such that the hierarchical relationship between the added sub-node and the child node is illustrated in the hierarchical tree.

52. (New) The computer readable medium as defined in Claim 19, wherein the method further comprises:

detecting that a pointer is over a portion of a displayed node in the hierarchical tree and that an input from a pointing device has been received such that the displayed node is selected; and

displaying a list of properties for the selected node in a separate pane of the graphical user interface in response to the selection of the node.

53. (New) The computer readable medium as defined in Claim 19, wherein the child node that is added is selected from a pre-programmed API function.

54. (New) The computer readable medium as defined in Claim 19, wherein the child node that is added is selected from a pre-programmed interface method.

55. (New) The system as defined in Claim 20, further comprising:

a module configured to receive a request to add a sub-node to the child node; and

a module configured to display a representation of the added sub-node such that the hierarchical relationship between the added sub-node and the child node is illustrated in the hierarchical tree.

56. (New) The system as defined in Claim 20, further comprising:

a module configured to detect that a pointer is over a portion of a displayed node in the hierarchical tree and that an input from a pointing device has been received such that the displayed node is selected; and

a module configured to display a list of properties for the selected node in a separate pane of the graphical user interface in response to the selection of the node.

57. (New) The system as defined in Claim 20, wherein the child node that is added is selected from a pre-programmed API function.

58. (New) The system as defined in Claim 20, wherein the child node that is added is selected from a pre-programmed interface method.